University of Stuttgart
Institute for Theoretical Chemistry

Use of automated term-generation and general tensor contractions in electronic structure theory

Andreas Köhn

ESNT-Workshop
“Automated tools for many-body theory”
June 6, 2023

Motivation

Main target: Electronic Schrödinger equation for many-electron systems (molecules)

$$\left(\hat{T}_\mathrm{e} + \hat{V}_\mathrm{ne} + \hat{V}_\mathrm{ee}\right)|\Psi_\mathrm{e}\rangle = |\Psi_\mathrm{e}\rangle E_\mathrm{e}$$

Main formalism: Second quantization

$$\hat{V}_\mathrm{ee} = \frac{1}{4}\sum_{pqrs} g_{pr}^{qs}\ \hat{a}^p \hat{a}^r \hat{a}_s \hat{a}_q$$

Main ansatz for wavefunctions: $e^{\hat{T}}|\Psi_\mathrm{ref}\rangle$

Main issue: lengthy expressions

$$\begin{aligned}
&\mathcal{L}_{\mathrm{SP}} + V_i^{r''}\bar{t}_a^i + \bar{B}_i^{a''}\bar{t}_a^i - X_j^a F_i^j \bar{t}_a^i + B_i^{a''}\bar{t}_a^i + F_j^{p''}\Lambda_i^a \mathcal{R}_{ap''}^{ib}\bar{t}_b^j - F_i^{p''}\Lambda_j^a \mathcal{R}_{ap''}^{ib}\bar{t}_b^j - \Lambda_j^a G_{ik}^{jp''}\mathcal{R}_{ap''}^{ib}\bar{t}_b^k + \Lambda_i^a V_a^{b''}\bar{t}_b^i - \Lambda \\
&+ \Lambda_j^b V_{ik}^{ja}\bar{t}_b^k\bar{t}_a^i - \Lambda_j^a V_i^b t_a^i\bar{t}_b^j - G_{ik}^{ap''}\Lambda_j^b t_a^j \mathcal{R}_{bp''}^{ic}\bar{t}_c^k + G_{ik}^{ap''} t_a^i \Lambda_j^b \mathcal{R}_{bp''}^{jc}\bar{t}_c^k - G_{ij}^{ap''} t_a^i \Lambda_k^b \mathcal{R}_{bp''}^{jc}\bar{t}_c^k + \Lambda_{ij}^{ab} C_{ab}^{ic}\bar{t}_c^j + \Lambda_{ik}^{ab}(\\
&- \Lambda_{ij}^{ab} G_{ka}^{ip''}\mathcal{R}_{bp''}^{jc}\bar{t}_c^k + \Lambda_{ij}^{ab} V_{ab}^{ic}\bar{t}_c^j - F_j^{p''}\Lambda_{ik}^{ab} t_b^j \mathcal{R}_{ap''}^{ic}\bar{t}_c^k - \Lambda_{jl}^{ab} G_{ik}^{jp''} t_a^k \mathcal{R}_{bp''}^{ic}\bar{t}_c^l - \Lambda_{ij}^{ab} G_{kl}^{jp''} t_b^k \mathcal{R}_{ap''}^{ic}\bar{t}_c^l + \Lambda_{ik}^{ab} V_{ja}^{ic} t_b^j\bar{t}_c^k \\
&+ \Lambda_{jk}^{ac} G_{ia}^{bp''} t_b^j \mathcal{R}_{cp''}^{id}\bar{t}_d^k - \Lambda_{ij}^{ab} G_{kb}^{cp''} t_c^j \mathcal{R}_{ap''}^{id}\bar{t}_d^k + \Lambda_{jk}^{bc} G_{ic}^{ap''} t_a^j \mathcal{R}_{bp''}^{jd}\bar{t}_d^k + \Lambda_{ik}^{ab} V_{jl}^{kc} t_{ab}^{il}\bar{t}_c^j - \Lambda_{ik}^{ab} V_j^c t_{ab}^{ij}\bar{t}_c^k + \Lambda_{kl}^{ab} V_{ij}^{kc} t_{ab}^{ij}\bar{t}_c^l \\
&- G_{kl}^{bp''}\Lambda_{ij}^{ac} t_{ab}^{ij}\mathcal{R}_{cp''}^{kd}\bar{t}_d^l + G_{il}^{ap''}\Lambda_{jk}^{bc} t_{ab}^{ij}\mathcal{R}_{cp''}^{kd}\bar{t}_d^l - G_{ik}^{ap''}\Lambda_{jl}^{bc} t_{ab}^{ij}\mathcal{R}_{cp''}^{kd}\bar{t}_d^l - G_{ij}^{ap''}\Lambda_{kl}^{bc} t_{ac}^{ij}\mathcal{R}_{bp''}^{kd}\bar{t}_d^l + \tfrac{1}{2}\Lambda_{il}^{ab} V_{jk}^{ic} t_a^j t_b^k\bar{t}_c^l \\
&- G_{jl}^{cp''}\Lambda_{ik}^{ab} t_b^j t_c^k \mathcal{R}_{ap''}^{id}\bar{t}_d^l - G_{ij}^{bp''}\Lambda_{kl}^{ac} t_a^j t_b^k \mathcal{R}_{cp''}^{id}\bar{t}_d^l - G_{ik}^{ap''}\Lambda_{jl}^{bc} t_c^k t_a^j \mathcal{R}_{bp''}^{id}\bar{t}_d^l + \bar{\Lambda}_i^a [V^\dagger]_a^i + \bar{\Lambda}_i^a \bar{B}_a^i - \bar{\Lambda}_i^a F_j^i X_a^j + \bar{\Lambda}_i^a \\
&+ \bar{\Lambda}_j^b F_k^j X_{ib}^{ka}\bar{t}_a^i + \bar{\Lambda}_{ja}^{ib} F_i^j \bar{t}_b^k - \bar{\Lambda}_i^a F_j^i X_a^b\bar{t}_b^j + \bar{\Lambda}_i^a B_a^i + \bar{\Lambda}_i^a B_a^{b''}\bar{t}_b^i - \bar{\Lambda}_i^a B_{ja}^{ib}\bar{t}_b^j - \bar{\Lambda}_j^a [\mathcal{R}^\dagger]_{ia}^{bp''} G_{kp''}^{ij} t_b^k + \bar{\Lambda}_i^a [V^\dagger]_a^b t_b^i \\
&+ \bar{\Lambda}_j^c [C^\dagger]_{ic}^{ab} t_{ab}^{ij} + \bar{\Lambda}_k^b [\mathcal{R}^\dagger]_{ib}^{cp''} G_{jp''}^{ia} t_{ac}^{jk} - \bar{\Lambda}_k^c [\mathcal{R}^\dagger]_{ic}^{ap''} G_{jp''}^{kb} t_{ab}^{ij} + \bar{\Lambda}_j^c [V^\dagger]_{ic}^{ab} t_{ab}^{ij} - \bar{\Lambda}_k^a [\mathcal{R}^\dagger]_{ia}^{bp''} G_{jp''}^{ic} t_b^j t_c^k + \bar{\Lambda}_i^a [\mathcal{R} \\
&+ \bar{\Lambda}_j^b [V^\dagger]_{ib}^{ac} t_c^j t_a^i + \bar{\Lambda}_k^c [\mathcal{R}^\dagger]_{jc}^{bp''} G_{ip''}^{ad} t_a^j t_{bd}^{ik} - \bar{\Lambda}_k^b [\mathcal{R}^\dagger]_{jb}^{dp''} G_{ip''}^{ac} t_c^j t_{ad}^{ik} + \bar{\Lambda}_k^a [\mathcal{R}^\dagger]_{ic}^{ap''} G_{jp''}^{bd} t_d^k t_{ab}^{ij} + \bar{\Lambda}_j^c [\mathcal{R}^\dagger]_{kc}^{dp''} G_{ip''}^{ab} t_
\end{aligned}$$

Generalized Contraction Code (GeCCo)

Text art by Mathias Pabst

First git log message: `Thu Mar 15 12:40:33 2007 +0100`

Designed based on ideas gathered from Jeppe Olsen's LUCIA code

Public version at `github.com/ak-ustutt/GeCCo-public`

Disclaimer: Pure development code with (unfortunately) bad documentation and a lot of hot-fixes 🥵

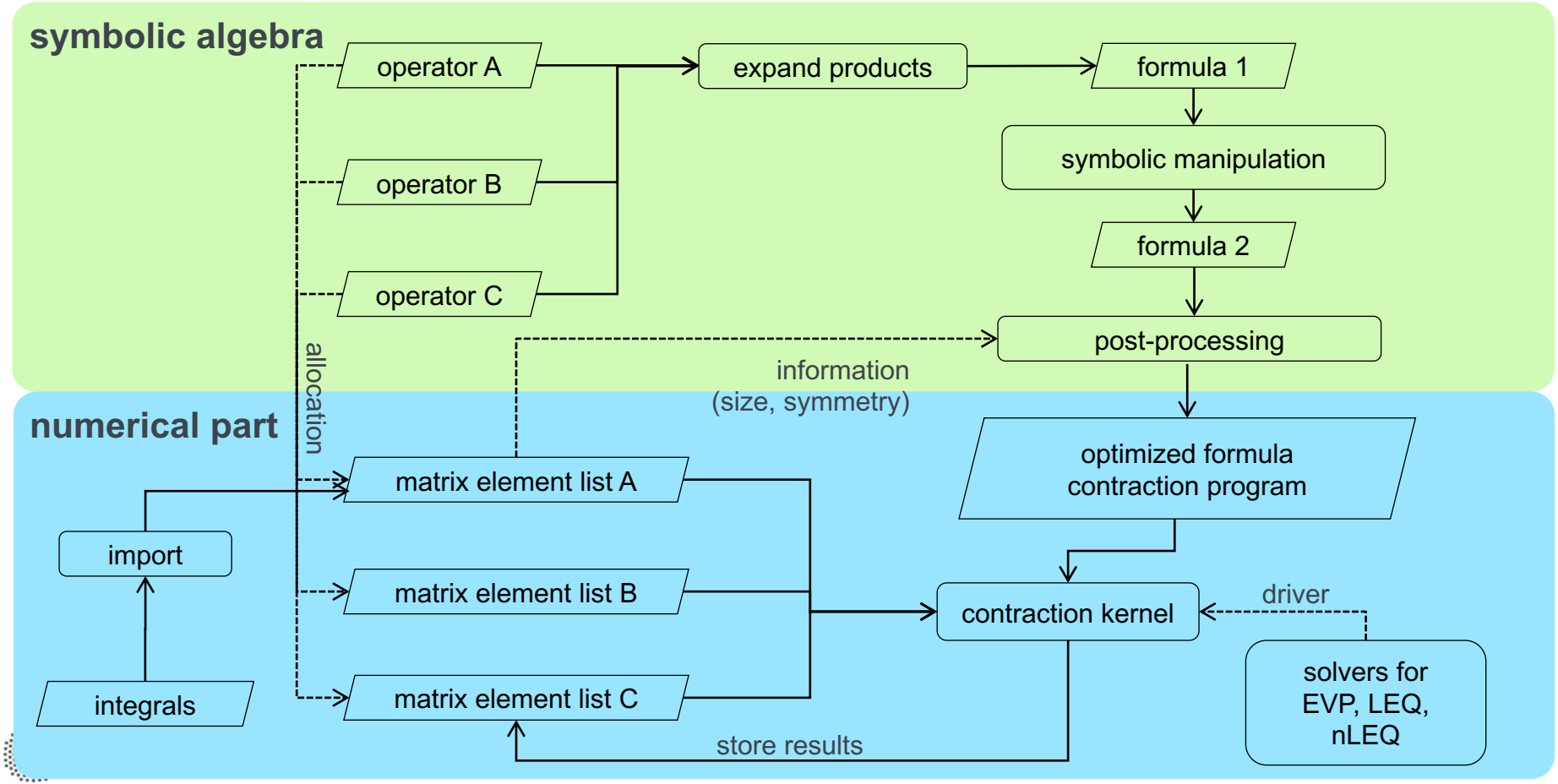
Mainly stand-alone code, integrals from Dalton, Molpro (and everything that provides a `FCIDUMP` file)

Originally intended for:	General-order CC (with selection)
But then used for:	Explicitly-correlated CC
And more lately:	Multireference CC

Contributors:

Gareth Richings
Matthias Hanauer
Pradipta Samantha
Yuri Aoto
Arne Bargholz
Josh Black

Organization of GeCCo – basic idea



Generating equations: Basic objects

Representation of operators:

$$\hat{T}_2 = \sum_{i,u,a>b} t_{ab}^{iu} \{\hat{a}^a \hat{a}^b \hat{a}_u \hat{a}_i\}$$

name: `T2`

H P V H P V

occupation: `(0,2,0;1,0,1)` or `"PP;HV"`

creations annihilations

$$\hat{H}:$$

$$E_\mathrm{i} = \langle 0|\hat{H}|0\rangle$$

$$\sum_{i,j} f_i^j \{\hat{a}^i \hat{a}_j\}$$

$$\sum_{i>j,u,a} g_{ij}^{au} \{\hat{a}^i \hat{a}^j \hat{a}_u \hat{a}_a\}$$

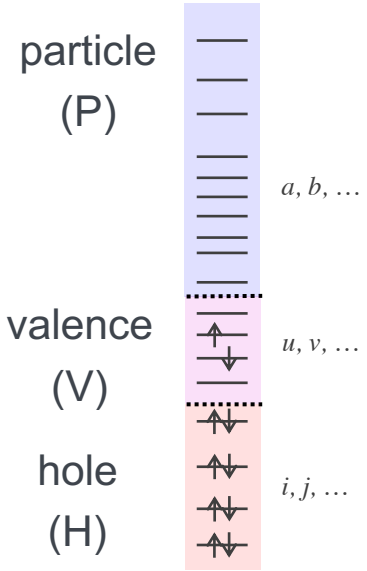
name: `H`

occupation: `(0,0,0;0,0,0)` or `";"`

occupation: `(1,0,0;1,0,0)` or `"H;H"`

occupation: `(2,0,0;0,1,1)` or `"HH;PV"`

“blocks”



inspiration: LUCIA (J. Olsen)

Generating equations: Basic objects

Representation of multiconfigurational state:

$$|\Psi_0\rangle = \sum_{tuvw} C^{(0)}_{tuvw} \{\hat{a}^t \hat{a}^u \hat{a}^v \hat{a}^w\} |0\rangle$$

name: `C0`

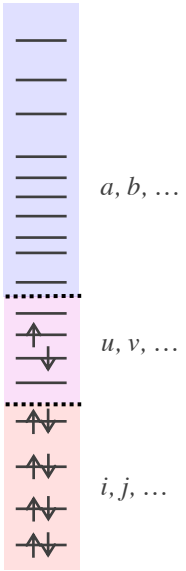
occupation: `(0,0,4;0,0,0)` or `"VVVV;"`

$$\langle\Psi_0| = \sum_{tuvw} \langle 0|\{\hat{a}_w \hat{a}_b \hat{a}_u \hat{a}_t\} C^{(0)}_{tuvw}$$

address as: `C0^+`

occupation (transposed):

`(0,0,0;0,0,4)` or `";VVVV"`



Representation of density matrices:

$$\gamma^{tuv}_{t'u'v'} = \langle\Psi_0|\{\hat{a}^t \hat{a}^u \hat{a}^v \hat{a}_{v'} \hat{a}_{u'} \hat{a}_{t'}\}|\Psi_0\rangle$$
$$= -\langle\Psi_0|\{\hat{a}_{v'} \hat{a}_{u'} \hat{a}_{t'} \hat{a}^t \hat{a}^u \hat{a}^v\}|\Psi_0\rangle$$

occupation:

`(0,0,0;0,0,3|0,0,3;0,0,0)`

or `";VVV|VVV;"`

”double-vertex repr.” (see later)

Generating equations: Operator products

Binary product:

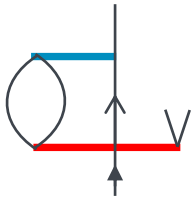
Wick

$$A^{cd}_{ia}\{\hat{a}^i\hat{a}^a\hat{a}_d\hat{a}_c\}B^{jkt}_{efg}\{\hat{a}^e\hat{a}^f\hat{a}^g\hat{a}_t\hat{a}_k\hat{a}_j\} = (C_0)^{jkcdt}_{iaefg}\{\hat{a}^i\hat{a}^a\hat{a}_d\hat{a}_c\hat{a}^e\hat{a}^f\hat{a}^g\hat{a}_t\hat{a}_k\hat{a}_j\} +$$

$$(C_{1,1})^{jkdt}_{iafg}\{\hat{a}^i\hat{a}^a\hat{a}_d\hat{a}_c\hat{a}^e\hat{a}^f\hat{a}^g\hat{a}_t\hat{a}_k\hat{a}_j\} + \ldots$$

$$(C_{3,1})^{jt}_{ag}\{\hat{a}^i\hat{a}^a\hat{a}_d\hat{a}_c\hat{a}^e\hat{a}^f\hat{a}^g\hat{a}_t\hat{a}_k\hat{a}_j\} + \ldots$$

(1,1,0;0,2,0) $[A]$ (0,3,0;2,1,0) $[B]$



Contraction matrix

represent as

	A	B	C
A	(0,0,0;0,0,0)	(1,0,0;0,2,0)	(0,1,0;0,0,0)
B	(0,2,0;1,0,0)	(0,0,0;0,0,0)	(0,1,0;1,0,1)

(0,2,0;1,0,1)

Unique rep. for all topologically equivalent contractions (diagrams) (related to adjacency matrix)

No sign determination at this point

Formalism:

$$[A] = [A]_c + [A]_0 \qquad [B]_c = [A^\dagger]_c$$

$$[B] = [B]_c + [B]_0 \qquad [C] = [A]_0 + [B]_0$$

inspiration: LUCIA (J. Olsen) and M. Kállay's publication on CC eq.s

Generating equations: Operator products

General product of *n* operators:

	A_1	A_2	…	A_n	R	R'
A_1	$[A_1]_1$	$[A_1]_2$	…	$[A_1]_n$	$[A_1]_0$	$[A_1]_{0'}$
A_2	$[A_2]_1$	$[A_2]_2$	…	$[A_2]_n$	$[A_2]_0$	$[A_2]_{0'}$
…	…	…	…	…	…	…
A_n	$[A_n]_1$	$[A_n]_2$	…	$[A_n]_n$	$[A_n]_0$	$[A_n]_{0'}$

> 1 result vertex, if no reordering desired, e.g. for densities:

$$\gamma^{ib}_{aj} \leftarrow \sum_{ck} \lambda^{cb}_{kj} t^{ik}_{ac}$$

Recursive algorithm to determine all possibilities (for given result occupation)

Most frequent case: `(0,0,0;0,0,0)` (i.e. scalar result, “closed diagrams”)

Store result as sequence of contraction matrices (= “formula”)

(+ info on num. prefactor)

Generating equations: Symbolic manipulations

Typical strategy:

Define method via stationary energy functional

$$\mathcal{L} = \langle \Phi_0 | (1 + \hat{\Lambda}) e^{-\hat{T}} \hat{H} e^{\hat{T}} | \Phi_0 \rangle$$

Get working equations as formal derivatives

$$\frac{\partial}{\partial \lambda_\rho} \mathcal{L} = \langle \Phi_0 | \hat{\tau}_\rho^\dagger e^{-\hat{T}} \hat{H} e^{\hat{T}} | \Phi_0 \rangle$$

$$\frac{\partial}{\partial t_\rho} \mathcal{L} = \langle \Phi_0 | (1 + \hat{\Lambda}) [e^{-\hat{T}} \hat{H} e^{\hat{T}}, \hat{\tau}_\rho] | \Phi_0 \rangle$$

Example term:

	LAM	H	T	L
LAM	(0,0,0;0,0,0)	(0,0,0;0,1,0)	(1,0,0;0,0,0)	(0,0,0;0,0,0)
H	(0,1,0;0,0,0)	(0,0,0;0,0,0)	(1,0,0;0,2,0)	(0,0,0;0,0,0)
T	(0,0,0;1,0,0)	(0,2,0;1,0,0)	(0,0,0;0,0,0)	(0,0,0;0,0,0)

	H	T	OMG
H	(0,0,0;0,0,0)	(1,0,0;0,2,0)	(0,1,0;0,0,0)
T	(0,2,0;1,0,0)	(0,0,0;0,0,0)	(0,0,0;1,0,0)

	LAM	H	JT
LAM	(0,0,0;0,0,0)	(0,0,0;0,1,0)	(1,0,0;0,0,0)
H	(0,1,0;0,0,0)	(0,0,0;0,0,0)	(1,0,0;0,2,0)

Generating equations: Further manipulations

- Factor out $R = LH + LHT$ $\xrightarrow{I = H + HT}$ $R = LI$
- Expand $R = LI$ $\xrightarrow{I = H + HT}$ $R = LH + LHT$
- Transpose $HT + T^\dagger HT$ $\longrightarrow$ $T^\dagger H + T^\dagger HT$
- Sum up $HT + T^\dagger HT - T^\dagger H - T^\dagger HT$ $\longrightarrow$ $HT - T^\dagger H$

Generating equations: Input language

`make` inspired work-flow driver:

define targets, dependencies, rules (= command sequences)

V0: hard-coded

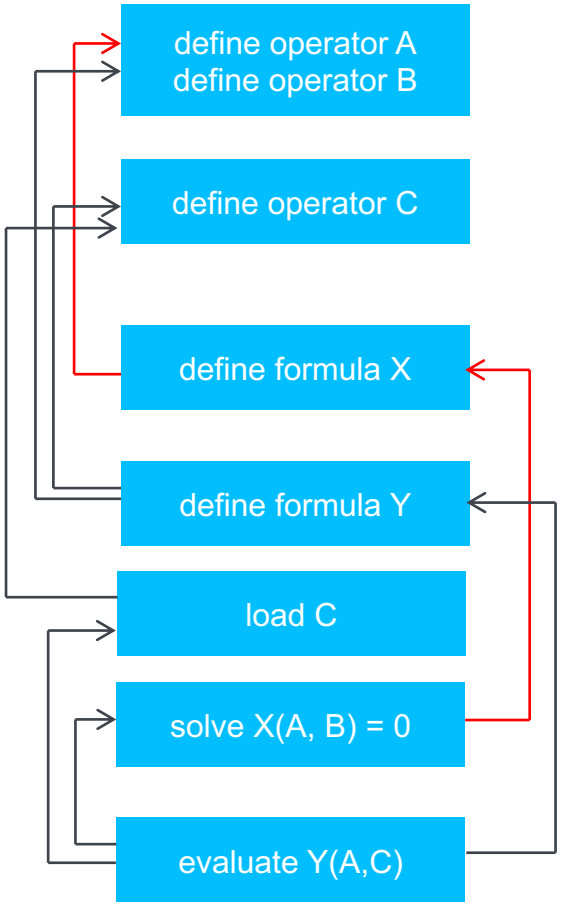
V1: simple self-written parser

V2: use Python as input parser (allow additional logic)

V2a: Python-based parsing of equation strings

Note: Python parsing only at initial input read-in

Conversion to input for Fortran, actual program is purely Fortan90 (and sometimes beyond)



Generating equations: Input language example

```
# Operators ------------------------------------
new_target('CCD_OPS')
DEF_SCALAR({LABEL:'LCCD'})
DEF_SCALAR({LABEL:'ECCD'})
DEF_EXCITATION({LABEL:'T2',MIN_RANK:2,MAX_RANK:2})
CLONE_OPERATOR({LABEL:'O2',TEMPLATE:'T2'})
CLONE_OPERATOR({LABEL:'L2',TEMPLATE:'T2',ADJOINT:True})

# CC Lagrangian --------------------------------
new_target('CCD_LAG')
depend('H')
depend('CCD_OPS')
form_l1 = stf.Formula("CCD_LAG:LCCD=<H+[H,T2]>")
form_l1.append("<L2*(H+[H,T2]+0.5*[[H,T2],T2])>")
form_l1.set_rule()
PRINT_FORMULA({LABEL:'CCD_LAG',MODE:'SHORT'})
```

$$\hat{T}_2$$

$$\Omega_2 = \langle \Phi_0 | \hat{a}_{ab}^{ij} \bar{H} | \Phi_0 \rangle$$

$$\hat{\Lambda}_2$$

$$\mathcal{L} = \langle \Phi_0 | (1 + \hat{\Lambda}_2) e^{\cdot \hat{T}_2} \hat{H} e^{\hat{T}_2} | \Phi_0 \rangle$$

$$= \langle \Phi_0 | \hat{H} + [\hat{H}, \hat{T}_2] | \Phi_0 \rangle$$

$$+ \langle \Phi_0 | \hat{\Lambda}_2 (\hat{H} + [\hat{H}, \hat{T}_2] + \frac{1}{2}[[\hat{H}, \hat{T}_2], \hat{T}_2]) | \Phi_0 \rangle$$

Generating equations: Input I

```
# Operators ----------------------------------
new_target('CCD_OPS')
DEF_SCALAR({LABEL:'LCCD'})
DEF_SCALAR({LABEL:'ECCD'})
DEF_EXCITATION({LABEL:'T2',MIN_RANK:2,MAX_RANK
CLONE_OPERATOR({LABEL:'O2',TEMPLATE:'T2'})
CLONE_OPERATOR({LABEL:'L2',TEMPLATE:'T2',ADJOI

# CC Lagrangian ------------------------------
new_target('CCD_LAG')
depend('H')
depend('CCD_OPS')
form_l1 = stf.Formula("CCD_LAG:LCCD=<H+[H,T2]>")
form_l1.append("<L2*(H+[H,T2]+0.5*[[H,T2],T2])>")
form_l1.set_rule()
PRINT_FORMULA({LABEL:'CCD_LAG',MODE:'SHORT'})

# CC E and Residual --------------------------
new_target('CCD_E_R')
depend('CCD_LAG')

DERIVATIVE({LABEL_RES:'CCD_R',LABEL_IN:'CCD_LAG',
            OP_RES:'O2',OP_DERIV:'L2'})
INVARIANT({LABEL_RES:'CCD_E',LABEL_IN:'CCD_LAG',
            OP_RES:'ECCD',OPERATORS:['L2']})
```

```
 1 LCCD(1) <-  1.000000 H(1)
 2 LCCD(1) <-  1.000000 H(8) T2(1) [1,2,HH,PP]
 3 LCCD(1) <-  1.000000 L2(1) H(12) [1,2,HH,PP]
 4 LCCD(1) <-  1.000000 L2(1) H(2) T2(1) [1,2,H,][1,3,H,PP][2,3,H,]
 5 LCCD(1) <-  1.000000 L2(1) H(5) T2(1) [1,2,,P][1,3,HH,P][2,3,,P]
 6 LCCD(1) <-  1.000000 L2(1) H(6) T2(1) [1,2,HH,][1,3,,PP][2,3,HH,]
 7 LCCD(1) <-  1.000000 L2(1) H(10) T2(1) [1,2,H,P][1,3,H,P][2,3,H,P]
 8 LCCD(1) <-  1.000000 L2(1) H(14) T2(1) [1,2,,PP][1,3,HH,][2,3,,PP]
 9 LCCD(1) <-  1.000000 L2(1) H(8) T2(1) T2(1) [1,3,H,][2,3,H,PP][1,4,H,PP][2,4,H,]
10 LCCD(1) <-  1.000000 L2(1) H(8) T2(1) T2(1) [1,3,HH,][2,3,,PP][1,4,,PP][2,4,HH,]
11 LCCD(1) <-  1.000000 L2(1) H(8) T2(1) T2(1) [1,3,,P][2,3,HH,P][1,4,HH,P][2,4,,P]
12 LCCD(1) <-  0.500000 L2(1) H(8) T2(1) T2(1) [1,3,H,P][2,3,H,P][1,4,H,P][2,4,H,P]
```

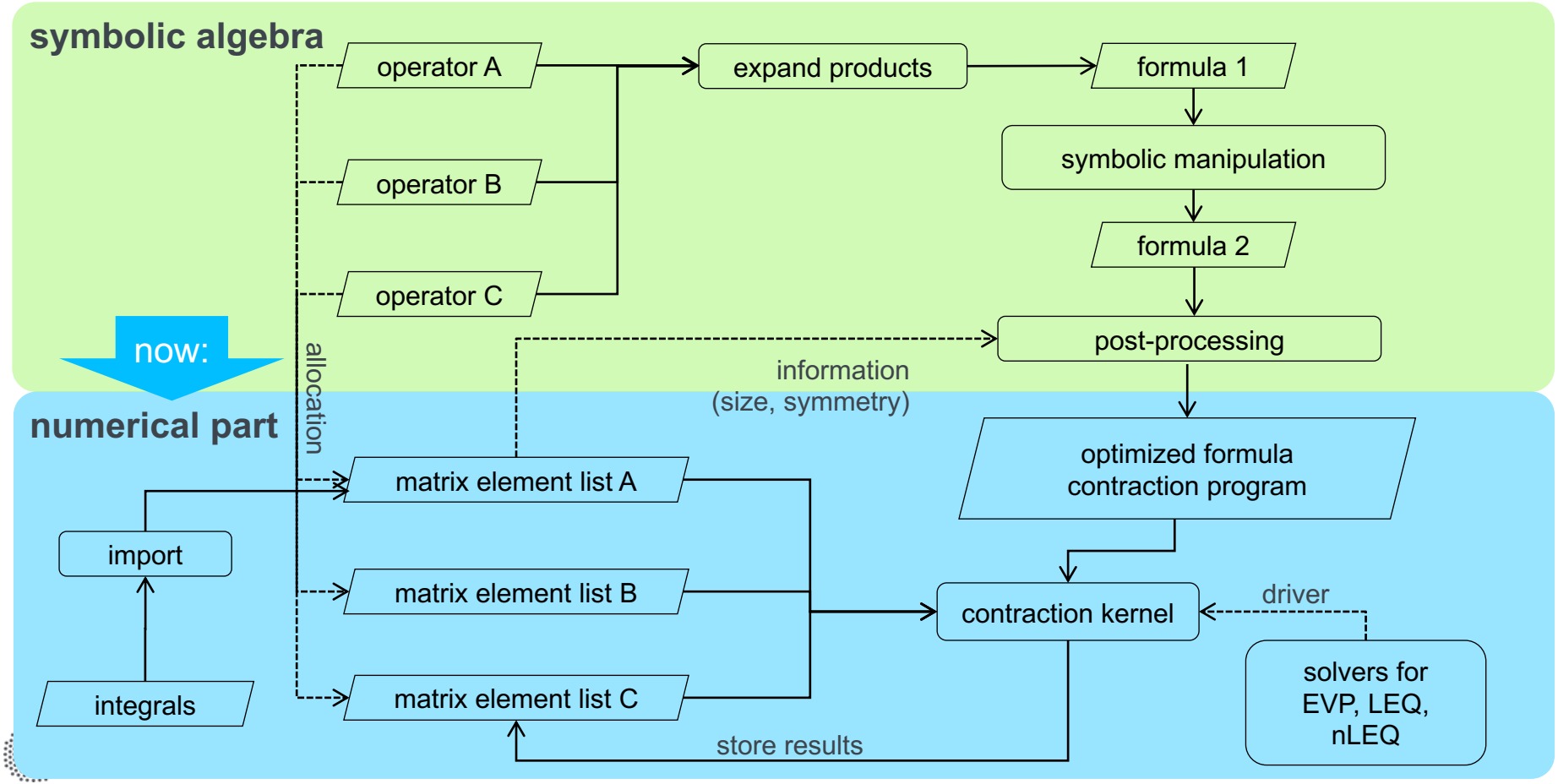
$$\mathcal{L} = \langle \Phi_0 | (1 + \hat{\Lambda}_2) e^{\cdot \hat{T}_2} \hat{H} e^{\hat{T}_2} | \Phi_0 \rangle$$

$$= \langle \Phi_0 | \hat{H} + [\hat{H}, \hat{T}_2] | \Phi_0 \rangle$$

$$+ \langle \Phi_0 | \hat{\Lambda}_2 (\hat{H} + [\hat{H}, \hat{T}_2] + \frac{1}{2}[[\hat{H}, \hat{T}_2], \hat{T}_2]) | \Phi_0 \rangle$$

$$(\Omega_2)^{ij}_{ab} = \frac{\partial \mathcal{L}}{\partial \lambda^{ij}_{ab}}$$

Organization of GeCCo – basic idea



Towards numerical evaluation: Optimizations

Strength reduction → binary contractions

$$\sum_{cdkl} g_{kl}^{cd} t_{ab}^{ik} t_{cd}^{jl} = \sum_{k} \left(\sum_{cdl} g_{kl}^{cd} t_{cd}^{jl} \right) t_{ab}^{ik}$$

Common subexpression identification (recursive intermediate fact.)

$$\sum_{k} f_{j}^{k} t_{ab}^{ik} + \sum_{k} \left(\sum_{cdl} g_{kl}^{cd} t_{cd}^{jl} \right) t_{ab}^{ik} = \sum_{k} \left(f_{j}^{k} + \sum_{cdl} g_{kl}^{cd} t_{cd}^{jl} \right) t_{ab}^{ik}$$

not coupled here

Alternative: Pre-define intermediates (not automated)

➡ Optimized list of binary contractions

Towards numerical evaluation: Tensor contraction kernel

external indices

$$R^{ij}_{ab} \leftarrow \sum_{kld} g^{jd}_{kl} t^{ikl}_{abd}$$

contraction indices

General scheme:

$$A(\mathcal{K}) \rightarrow A(\mathcal{C}, \mathcal{K}_0)$$
$$B(\mathcal{L}) \rightarrow B(\mathcal{C}^\dagger, \mathcal{L}_0)$$
$$R(\mathcal{K}_0, \mathcal{L}_0) = A(\mathcal{C}, \mathcal{K}_0) B(\mathcal{C}^\dagger, \mathcal{L}_0)$$
$$R(\mathcal{K}_0, \mathcal{L}_0) \rightarrow R(\mathcal{I})$$

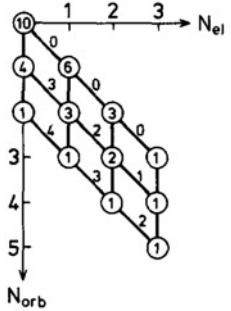
Address string is composed of substrings:

$$\mathcal{K} = \mathcal{K}^{\mathrm{c}}_{\mathrm{H}} \mathcal{K}^{\mathrm{c}}_{\mathrm{P}} \mathcal{K}^{\mathrm{c}}_{\mathrm{V}} \mathcal{K}^{\mathrm{a}}_{\mathrm{H}} \mathcal{K}^{\mathrm{a}}_{\mathrm{P}} \mathcal{K}^{\mathrm{a}}_{\mathrm{V}}$$

Generate unique flat index for substrings:

String resolutions come with sign changes and increased sparsity (unpacking):

$$(ij) \rightarrow \begin{matrix} 0 & + & + & + \\ - & 0 & + & + \\ - & - & 0 & + \\ - & - & - & 0 \end{matrix}$$



Batched unpacking, use sparsity in outer loops, innermost loops use `dgemm`

Olsen, Roos, Jørgensen, Jensen, JCP **89**, 2185 (1988)

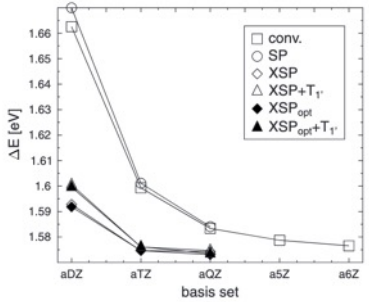
Kállay, Surján, JCP **115**, 2945 (2001)

Application examples

F12 theory

Complete basis correction

$$\exp(\hat{T}_1 + \hat{T}_2 + \hat{T}_{2'})|\Phi_0\rangle \qquad \hat{T}_{2'} = \sum_{i>j,k>l,\alpha>\beta} c_{kl}^{ij} \langle \alpha\beta|\hat{Q}_{12} f(r_{12})|kl\rangle \hat{a}_{kl}^{\alpha\beta}$$



Comment on Quintuple-ζ quality coupled-cluster correlation energies with triple-ζ basis sets by D. P. Tew, W. Klopper, C. Neiss and C. Hättig, *Phys. Chem. Chem. Phys.*, 2007, 9, 1921 [erratum]

David P. Tew,[a] Wim Klopper,*[a] Christian Neiss[b] and Christof Hättig[c]

Received 7th July 2008, Accepted 18th July 2008
First published as an Advance Article on the web 15th September 2008
DOI: 10.1039/b811567b

erroneous values. Our current implementation is in complete agreement with an independently programmed CCSD(F12) program that uses generalised contraction routines[2] and we are certain that the values reported here are reliable. The

2 A. Köhn and G. Richings, private communication.

$$\exp(\hat{T}_1 + \hat{T}_2 + \hat{R} + [\hat{R}, \hat{T}_1 + \hat{T}_2])|\Phi_0\rangle$$

"Full" CCSD-F12

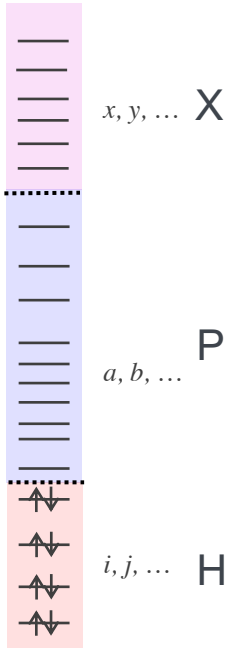
AK, Richings, Tew, JCP 129, 201103 (2008)
Shiozaki, Kamiya, Hirata, Valeev, JCP 129, 071101 (2008)
AK, Tew, JCP 133, 174117 (2010)

Explicitly-correlated triple excitations

AK, JCP 130, 131101 (2009)
AK, JCP 133, 174118 (2010)

CCSD-F12 response theory

AK, JCP 130, 104104 (2009)
Hanauer, AK, JCP 131, 124118 (2009)



Application examples

Multireference coupled-cluster theory

$$|\Psi\rangle = e^{\hat{T}}|\psi_0\rangle = e^{\hat{T}} \sum_{\mu} |\Phi_\mu\rangle c_\mu$$

$$|\Psi_0\rangle = \sum_{tuvw} C^{(0)}_{tuvw} \{\hat{a}^t \hat{a}^u \hat{a}^v \hat{a}^w\} |0\rangle$$

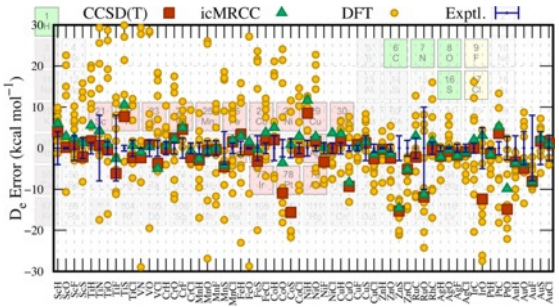
Evangelista, Gauss, JCP 134, 114102 (2011)
Hanauer, Köhn, JCP 134, 204111 (2011)

Interesting features:

Spin-adapted, strong analogy to single-reference theory, orbital invariance

But: Very complex eq.s

Benchmarking transition metal bond energies



Aoto, Lima Batista, AK, Oliveira-Filho, JCTC 13, 5291 (2017)

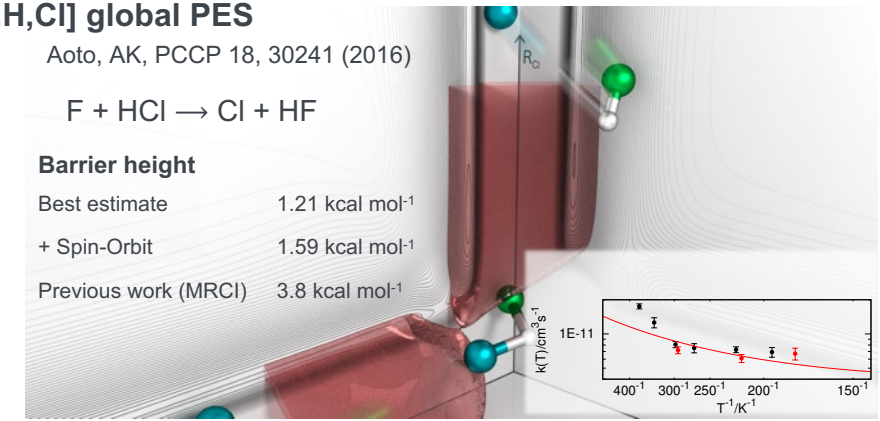
[F,H,Cl] global PES

Aoto, AK, PCCP 18, 30241 (2016)

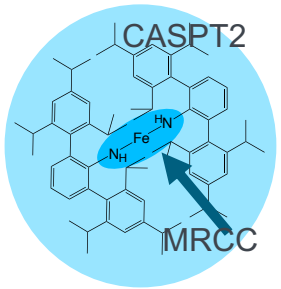
F + HCl ⟶ Cl + HF

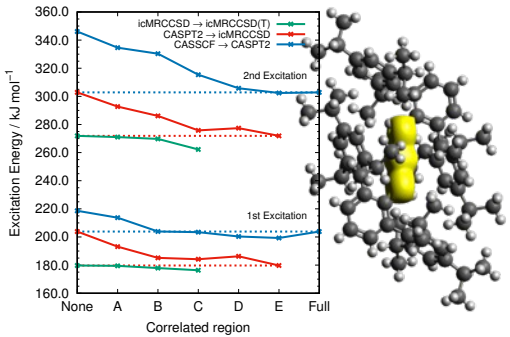
Barrier height

Best estimate	1.21 kcal mol^{-1}
+ Spin-Orbit	1.59 kcal mol^{-1}
Previous work (MRCI)	3.8 kcal mol^{-1}



Larger systems by embedding





Coughtrie, Giereth, Kats, Werner, AK, JCTC 14, 693 (2018)

Improving numerical evaluation: Translation

GeCCo's numerics are not bad, but certainly not good enough for routine applications

Couple to a more efficient tensor evaluation framework: ITF (Knizia, Shamasundar, Kats)

Encode algorithm in meta-language

```
tensor: R[aaii], !Create{type:disk; sym:01/23}, ResT2

. . .

for [k,j]:
   .R[abkj] += K[abkj]

   load K4E[abkj]
   .R[abkj] += K4E[abkj]
   drop K4E[abkj]

   alloc C[abkj]
   .C[abkj] += T[abkj]
   .C[abkj] += t[ak] t[bj]
   .R[abmn] += C[abkj] A[mnkj]
   drop C[abkj]
```

Hand-written code
Some optimization by Ch. Köppl, PhD-thesis

Uses intrinsic features of Molpro

integral-direct two-electron transformations

$$\frac{1}{2}\sum_{cd} g_{ab}^{cd}\left(t_{cd}^{ij} + t_c^i t_d^j\right)$$

coupling coefficients and densities for internally contracted theories

Shamasundar, Knizia, Werner, JCP **135**, 054101 (2011)
Kats, Manby, JCP **138**, 144101 (2013)

Improving numerical evaluation: Translation

Use pre-optimized sequence of binary contractions from GeCCO, translate to ITF code

Josh A. Black

Main issues: GeCCo is spin-orbital string based

- Must work out spin summation and explicit labels + signs
- Post-processing to meet ITF syntax (declarations etc.)
- Write new C++ driver code

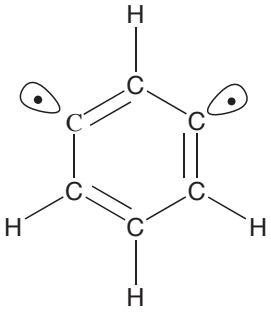
$$\sum_{vw} \gamma_w^v \sum_{abi} g_{iv}^{ab} t_{ab}^{iw}$$

$$\sum_{vw} \Gamma_w^v \sum_{abi} K_{iv}^{ab} (2T_{ab}^{iw} - T_{ba}^{iw})$$

Current status: Only CAS(2,2), only partial spin summation and no optimal factorization of non-linear terms

Black, Waigum, Adam, Shamasundar, AK, JCP **158**, 134801 (2023)

Performance



singlet state
CAS(2,2)

Approximation with linearization of “difficult terms” (many active space indices)

Parallelization: Too granular code currently, embarrassingly parallel tasks (Gradients, Hessian) work well

integral trafo + resort

Method (Impl.)	Basis set	Overhead [s]	Time/iter [s]	# iter
icMRCCSD (GeCCo)	cc-pVDZ	56	42	20
	cc-pVTZ	1282	441	21
icMRCCSD (ITF)	cc-pVDZ	0.7	10	21
	cc-pVTZ	3	68	20
	cc-pV5Z	238	3872	20
icMRCC/CEPA (ITF)	cc-pVTZ	3	5	31
	cc-pV5Z	237	251	31
CCSD (Molpro)	c-pVTZ	1	1.7	16
	cc-pV5Z	190	142	16

#bf: 104 (cc-pVDZ), 236 (cc-pVTZ), 766 (cc-pV5Z)

Black, Waigum, Adam, Shamasundar, AK, JCP **158**, 134801 (2023)

Future aims

GeCCo → Lizard `github.com/KoehnLab/Lizard`

Goals: Term generation features of GeCCo + code generation for tensor engines

Robert Adam

More modularity (or: less monolithic)

(interfaces to other term generators,
other tensor contraction kernels)

Python interface for input control and fast prototyping

Improvements, like:

- Generalization of index spaces (e.g. for tensor decompositions)
- More global representation of terms (expression trees)
- New treatment of index permutations

Future aims

In fact there are many related projects (Quantum Chemistry focus, as collected by Robert):

- Early work by Janssen & Schaefer [1]
- TCE (Hirata et al. [2])
- SMITH (Shiozaki et al. [3])
- ACE (Song et al. [4])
- MRCC (Kállay et al. [5])
- Orca-AGE (Krupika et al. [6])
- GeCCo (Köhn et al. [7])
- SeQuant (by Valeev group)
- drudge (Zhao [8])
- Cadabra (Peeters [9]; v2: Peeters [10])
- Datta & Gauss [11]
- MacLeod & Shiozaki [12]
- ITF (Molpro [13])
- squant (Kats)
- SIAL (ACES III [14])
- CTF (Solomonik et al. [15])
- libtensor (Epifanovsky et al. [16])
- tiledarray (Peng et al. [17])

Future aims

Brainstorming: How could we all better interact?

Some standard(s) to store generated equations?

Some API standard(s) for tensor contraction engines?

Review paper on best practices?

Some references for slide 24

1. Janssen, C. L. & Schaefer, H. F. The automated solution of second quantization equations with applications to the coupled cluster approach. en. Theor. Chim. acta 79, 1–42.. https://doi.org/10.1007/BF01113327 (1991).

2. Auer, A. A. et al. Automatic code generation for many-body electronic structure methods: the tensor contraction engine. Mol. Phys. 104, 211–228. https://doi.org/10.1080/00268970500275780 (2006)

3. Shiozaki, T., Kamiya, M., Hirata, S. & Valeev, E. F. Equations of explicitly-correlated coupled-cluster methods. en. Phys. Chem. Chem. Phys. 10, 3358–3370. https://pubs.rsc.org/en/content/articlelanding/2008/cp/b803704n (2008).

4. Song, C., Wang, L.-P. & Martínez, T. J. Automated Code Engine for Graphical Processing Units: Application to the Eective Core Potential Integrals and Gradients. J. Chem. Theory Comput. 12, 92–106. ISSN: 1549-9618. https://doi.org/10.1021/acs.jctc.5b00790 (2016)

5. Kállay, M. et al. The MRCC program system: Accurate quantum chemistry from water to proteins. J. Chem. Phys. 152, 074107.. https://aip.scitation.org/doi/10.1063/1.5142048 (2020).

6. Krupika, M., Sivalingam, K., Huntington, L., Auer, A. A. & Neese, F. A toolchain for the automatic generation of computer codes for correlated wavefunction calculations. J. Comput. Chem. 38, 1853–1868. https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.24833 (2017).

7. Köhn, A., Richings, G. W. & Tew, D. P. Implementation of the full explicitly correlated coupled-cluster singles and doubles model CCSD-F12 with optimally reduced auxiliary basis dependence. J. Chem. Phys. 129, 201103. https://aip.scitation.org/doi/10.1063/1.3028546 (2008).

8. Zhao, J. Symbolic solution for computational quantum many-body theory development. PhD thesis (Rice University, 2018).

https://hdl.handle.net/1911/105668

9. Peeters, K. Cadabra: a field-theory motivated symbolic computer algebra system. Comput. Phys. Commun. 176, 550–558. https://www.sciencedirect.com/science/article/pii/S0010465507000318 (2007).

Some references for slide 24 (cont.d)

10. Peeters, K. Cadabra2: computer algebra for field theory revisited. Journal of Open Source Soare 3, 1118 (2018).

11. Datta, D. & Gauss, J. A Non-antisymmetric Tensor Contraction Engine for the Automated Implementation of Spin-Adapted Coupled Cluster Approaches. J. Chem. Theory Comput. 9, 2639–2653. https://doi.org/10.1021/ct400216h (2013)

12. MacLeod, M. K. & Shiozaki, T. Communication: Automatic code generation enables nuclear gradient computations for fully internally contracted multireference theory. J. Chem. Phys. 142, 051103. https://aip.scitation.org/doi/10.1063/1.4907717 (2015).

13. Werner, H.-J. et al. The Molpro quantum chemistry package. J. Chem. Phys. 152, 144107. ISSN: 0021-9606. https://aip.scitation.org/doi/10.1063/5.0005081 (2020).

14. Deumens, E. et al. Soare design of ACES III with the super instruction architecture. WIREs Comput. Mol. Sci. 1, 895–901. https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.77 (2011)

15. Solomonik, E., Matthews, D., Hammond, J. R., Stanton, J. F. & Demmel, J. A massively parallel tensor contraction framework for coupled-cluster computations. J. Parallel Distr. Com. Domain-Specific Languages and High-Level Frameworks for High-Performance Computing 74, 3176–3190. https://www.sciencedirect.com/science/article/pii/S074373151400104X (2014)

16. Epifanovsky, E. et al. New implementation of high-level correlated methods using a general block tensor library for high-performance electronic structure calculations. J. Comput. Chem. 34, 2293–2309. https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.23377 (2013).

17. Peng, C., Calvin, J. A., Pavoevi, F., Zhang, J. & Valeev, E. F. Massively Parallel Implementation of Explicitly Correlated Coupled-Cluster Singles and Doubles Using TiledArray Framework. J. Phys. Chem. A 120, 10231–10244. https://doi.org/10.1021/acs.jpca.6b10150 (2016)